Title: Systems and Methods Extending An Existing Programming Language With Constructs

Docket No.: BEAS-01389US2

Inventor(s): Pal Takacsi-Nagy & Michael Douglas Blow Attorney: Sheldon R. Meyer Phone: (415) 362-3800 Serial No.: Confirm No: Filed: HEREWITH Sheet 1 of 3

100

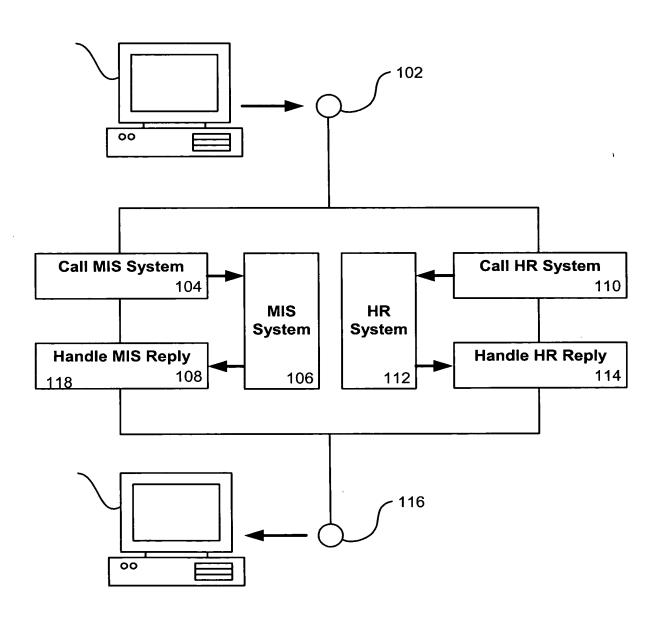


Figure 1

```
Title: Systems and Methods Extending An Existing Programming
Language With Constructs

Docket No.: BEAS-01389US2

Inventor(s): Pal Takacsi-Nagy & Michael Douglas Blow
Serial No.: Attorney: Sheldon R. Meyer
Confirm No: Phone: (415) 362-3800

Filed: HEREWITH Sheet 2 of 3
```

```
@jwf:flow flow ::
 cess>
  <receive name="Receive employee info" method="start"/>
  <parallel join-condition="AND">
    <br/>branch>
     <perform name="Request e-mail" method="reqMail"/>
     <receive name="Get e-mail" method="MIS mailReady"/>
     <perform name="Notify client about e-mail"</pre>
         method="mailNotification"/>
    </branch>
    <br/>branch>
     <perform name="Request benefits" method="reqBenefits"/>
     <receive name="Get benefits" method="hr benefitsReady"/>
     <perform name="Notify client about benefits"</pre>
         method="benefitsNotification"/>
    </branch>
  </parallel> * <perform name="Reply to requestor" method="end"/>
* </process>
```

Figure 2

```
Title: Systems and Methods Extending An Existing Programming
Language With Constructs

Docket No.: BEAS-01389US2
Inventor(s): Pal Takacsi-Nagy & Michael Douglas Blow
Serial No.: Attorney: Sheldon R. Meyer
Confirm No: Phone: (415) 362-3800
Filed: HEREWITH Sheet 3 of 3
```

```
@jwf:flow flow::
  corder = "PurchaseOrder">
    <receive name="Receive PO" method="getPO"/>
     <forEach name="processLineItems" var="lineitem"</pre>
          expression="getLineItems"
          parameters="inputPO">
     <perform name="Process line item" method="processOrder"/>
         <receive name="Handle service ack."
          method="orderService_sendAck"/>
    <perform name="Send reply to the client" method="sendReply"/>
  </process>
* xquery::
 define function getLineItems (element $po) returns element* {
    $po/DATAAREA/PROCESS_PO/POORDERLIN }
  define function concat (element $x1, element $x2) returns element {
    x1 + x2
  define function buildReply (element $x1) returns element {
     <reply>$x1</reply> }
public class PurchaseOrder {
  public XML inputPO;
  public XML lineitem;
    @jwf:xml-sequence
  public XML poAckList;
   * @jwf:transforms
  PoTransforms transforms;
    @jws:control
  public OrderProcessor orderService;
  void getPO(XML po) {
    inputPo = po;
  public void processOrder() {
    orderService.processOrder(lineItem);
  public void orderService_sendAck(XML ackedLine)
    throws Exception {
    poAckList = transforms.concat(poAckList, ackedLine);
  public void sendReply() {
    callback.reply(transforms.buildReply(poAckList));
  public Callback callback;
    public interface Callback {
     public void reply(XML ack);
}
```

Figure 3